Code-based Cryptography

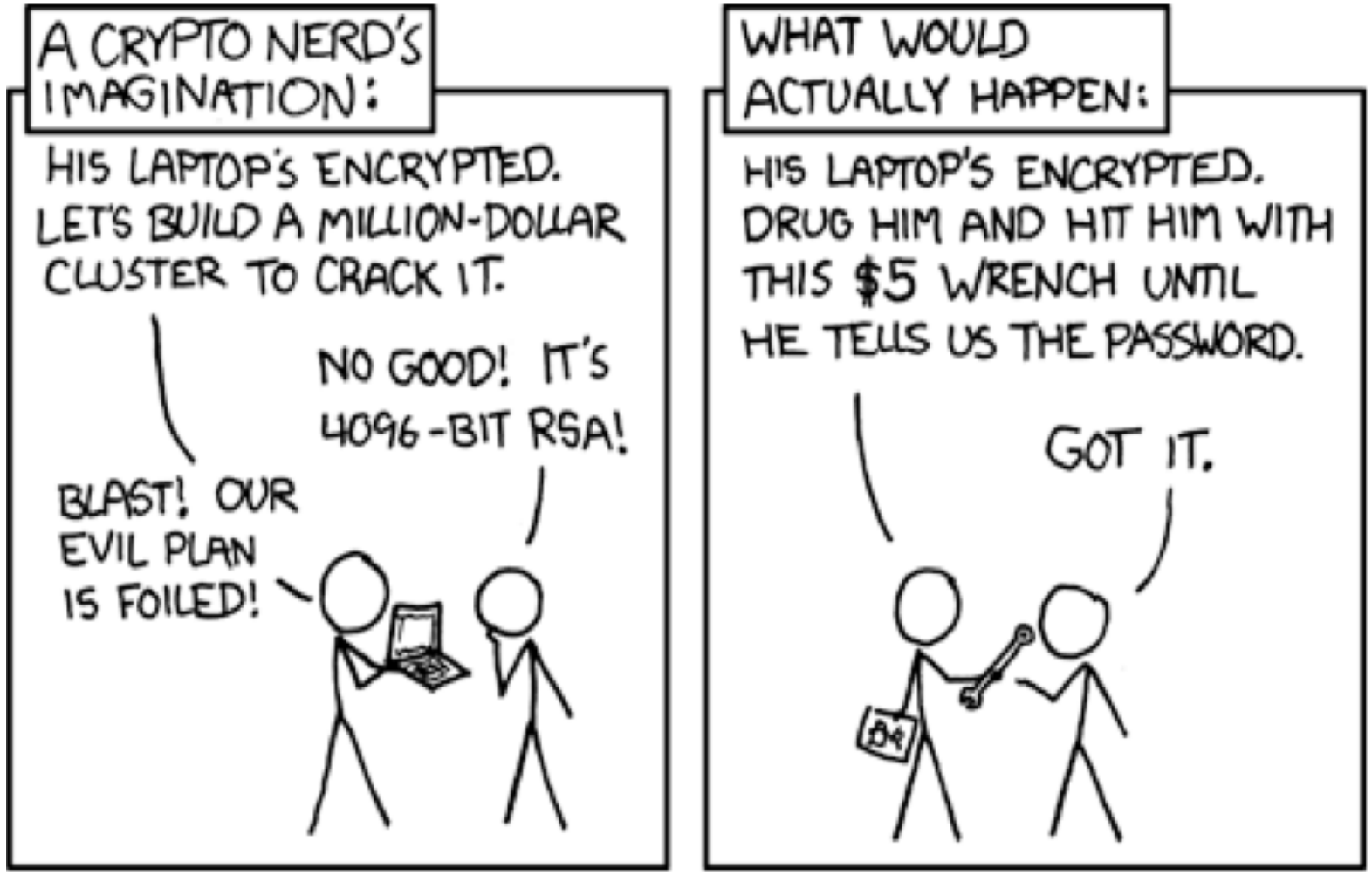
Side-Channel Attacks

Fraunhofer Workshop Series 01 – Post-Quantum Cryptography in Practice

Speaker: Dr. Bernhard Jungk

Fraunhofer SINGAPORE

Introduction to Side Channel Attacks


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Side-Channel Attacks

Background on Side-Channels

Timing Side-Channels

- Remote key extraction, many examples:
  - » 2005 "Cache-Timing Attack on AES", Bernstein
  - » 2017 "A Microarchitectural Side Channel Attack on Several Real-World Applications of Curve25519", Genkin et al.
- Meltdown and Spectre 
  - » Most processors vulnerable to one of the problem
  - » Extraction of kernel space memory possible
  - » Architectural problem of the processor designs
- Very high impact on most systems with remote interfaces (from smartcards to servers)

Side-Channel Attacks

Background on Side-Channels

Power/EM Side-Channel Attacks

- First published by Kocher in 1999
- Many variants
  - » Simple power analysis → Direct interpretation of traces
  - » Differential/correlation power analysis → Statistical modelling
  - » Template attacks → Profiling of device
  - » Machine-learning based approaches → Advanced profiling of device
- Countermeasures are costly
  - » Masking removes some leakages completely in theory
  - » Hiding reduces the signal to noise ratio
- High impact on embedded systems and smartcards

Side-Channel Attacks

Power and EM analysis – A short introduction

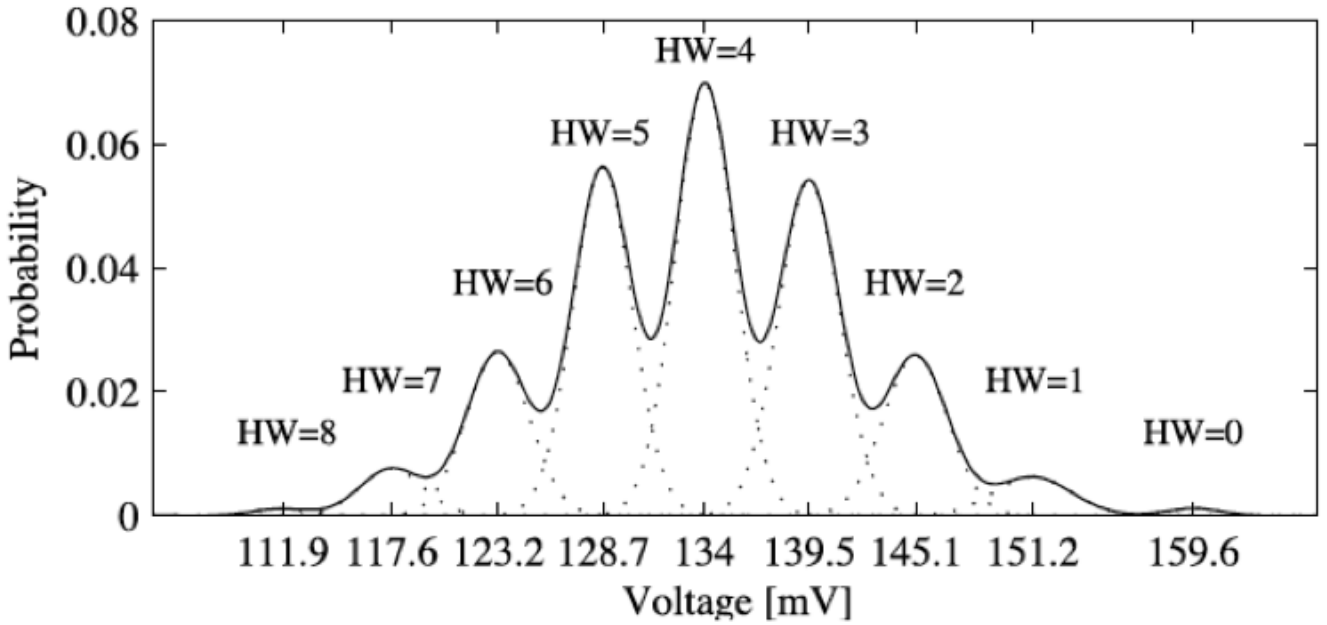
Energy consumption changes:

- Depending on the algorithm:
  - Different operations consume a different amount of energy.
- Depending on the input data:
  - In traditional CMOS switching consumes the highest amount of energy
  - Switching from 0 to 1 (or 1 to 0) consumes more energy than the 0 to 0 (or 1 to 1) transitions.

Side-Channel Attacks

Power and EM analysis – A short introduction

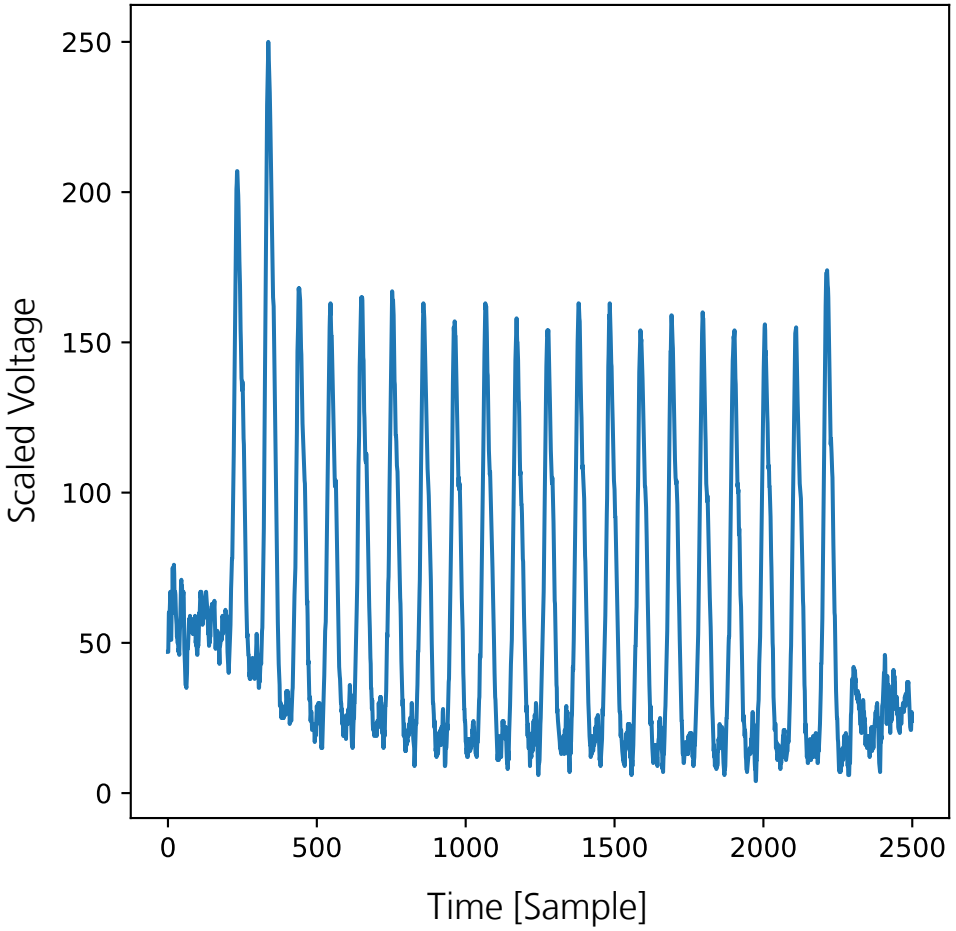
Statistical Modelling of Leakage



Fraunhofer SINGAPORE

Side-Channel Attacks

Power and EM analysis – A short introduction



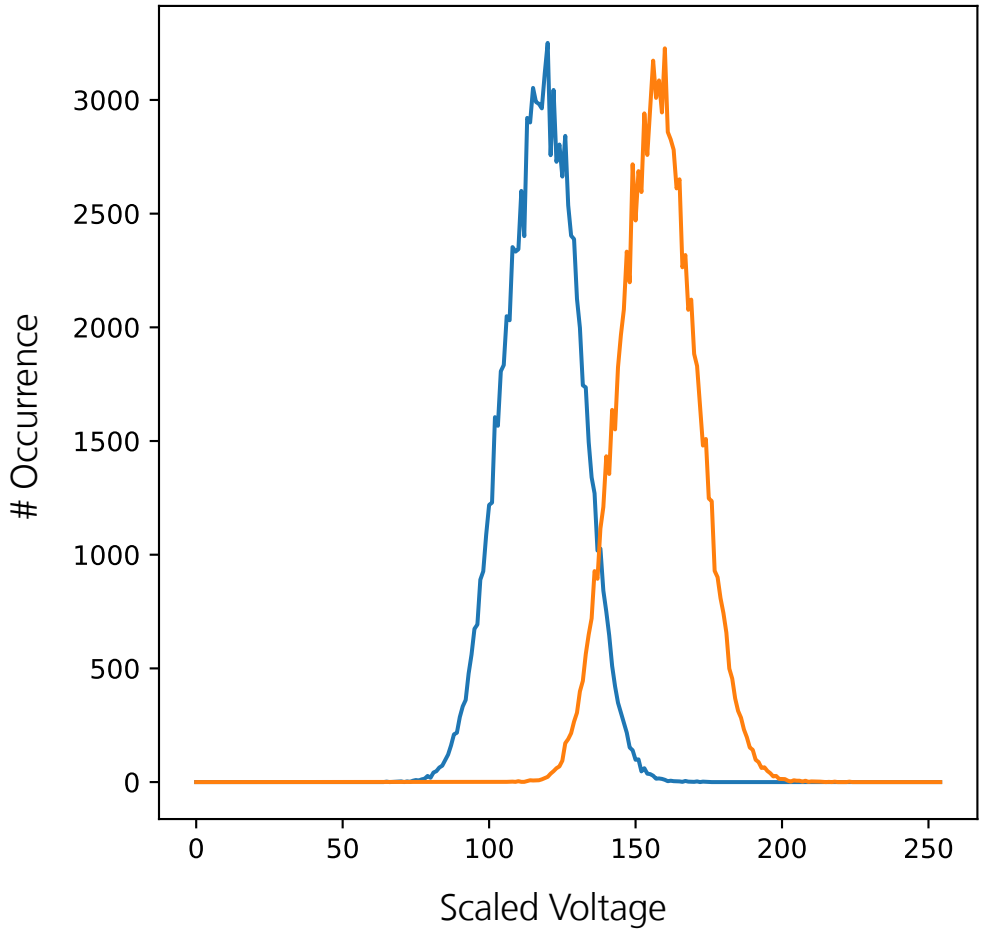
FPGA Implementation of ChaCha20:

- 20 clearly identifiable rounds
- Measured power consumption changes in each round:
  - Internal data
  - Measurement noise

Fraunhofer SINGAPORE

Side-Channel Attacks

Power and EM analysis – Modular Addition



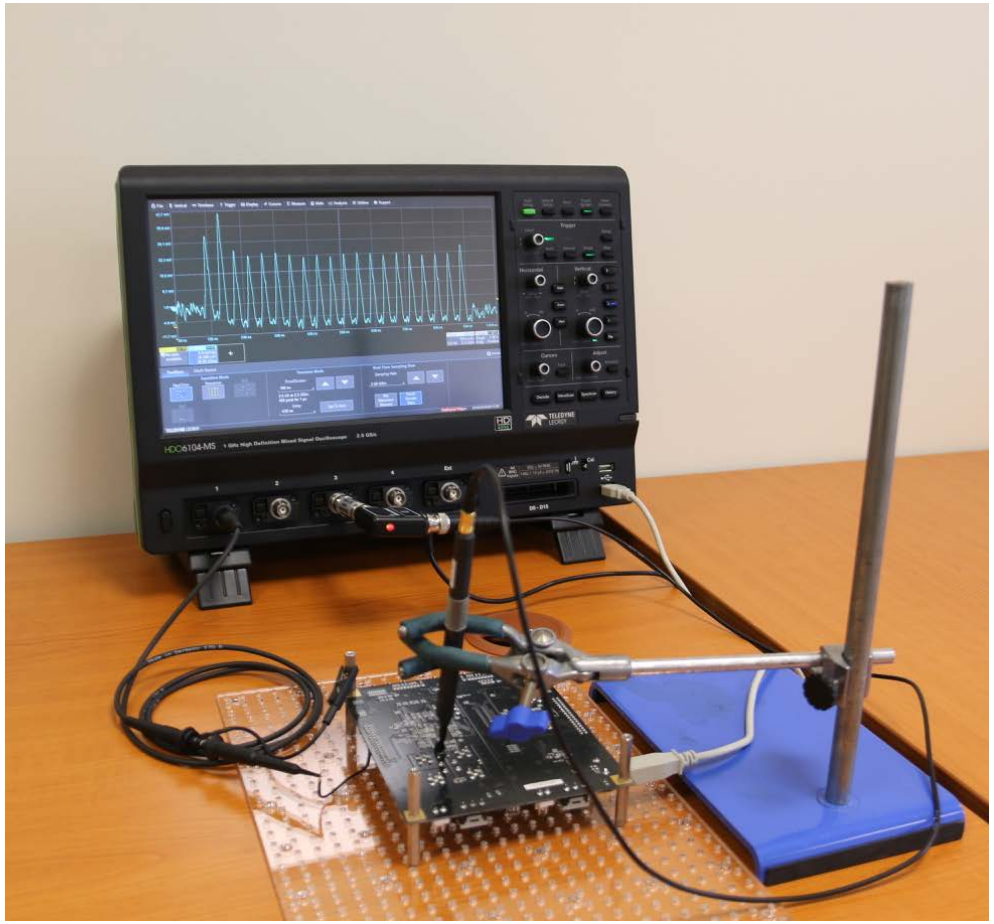
Protected FPGA Implementation of Modular Addition:

- PRNG for mask generation was deactivated
- Histogram of a leaking point of interest
- Different inputs clearly identifiable in histograms

Fraunhofer SINGAPORE

Side-Channel Attacks

Power and EM analysis – General Approach for Profiled Attack



Our setup for this attack:

Oscilloscope: Lecroy HDO6104-MS

Target: SASEBO GII

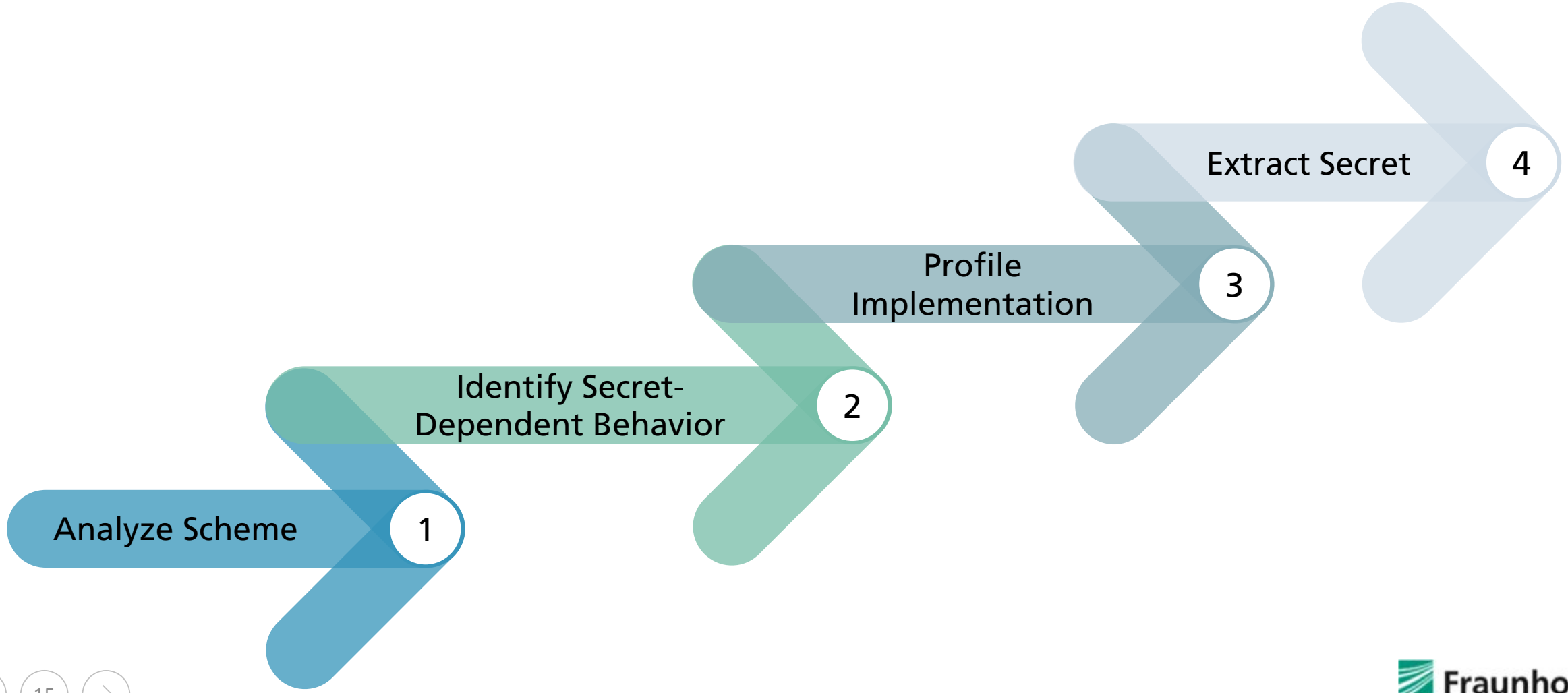
(Side-Channel Evaluation Board with Xilinx Virtex-5 FPGA)

Probe: Langer RF-U 5-2

Side Channel Attacks on the Niederreiter Cryptosystem

Attack Strategy

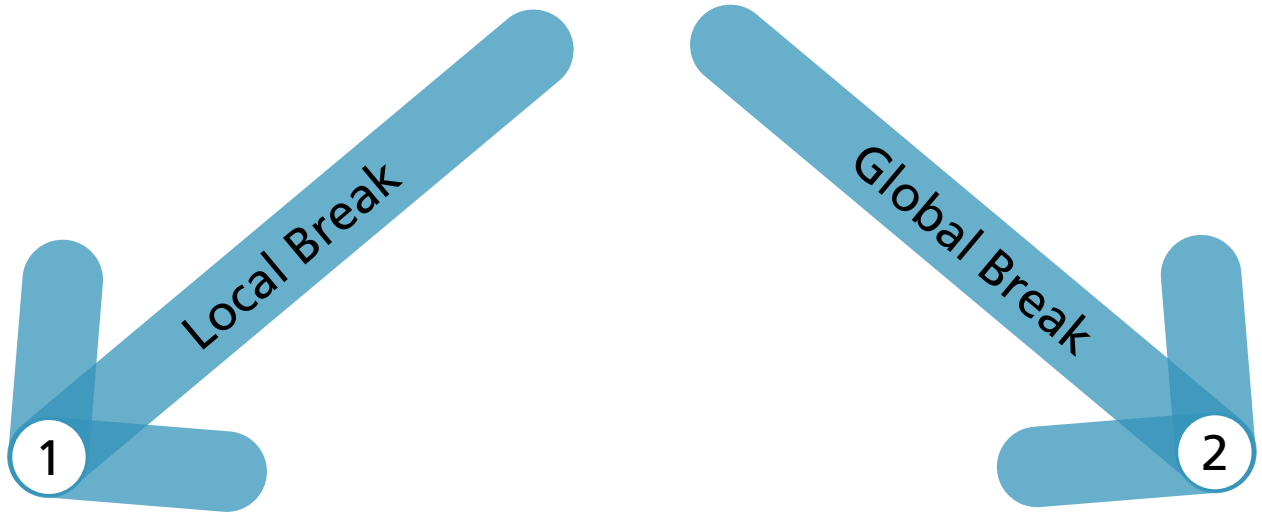
Niederreiter Cryptosystem



Fraunhofer SINGAPORE

Application to Code-Based Cryptography

Niederreiter Cryptosystem - Analysis



Decrypt a single ciphertext

Exploit properties of decryption

Extract private key

Exploit properties of decryption or key generation

Application to Code-Based Cryptography

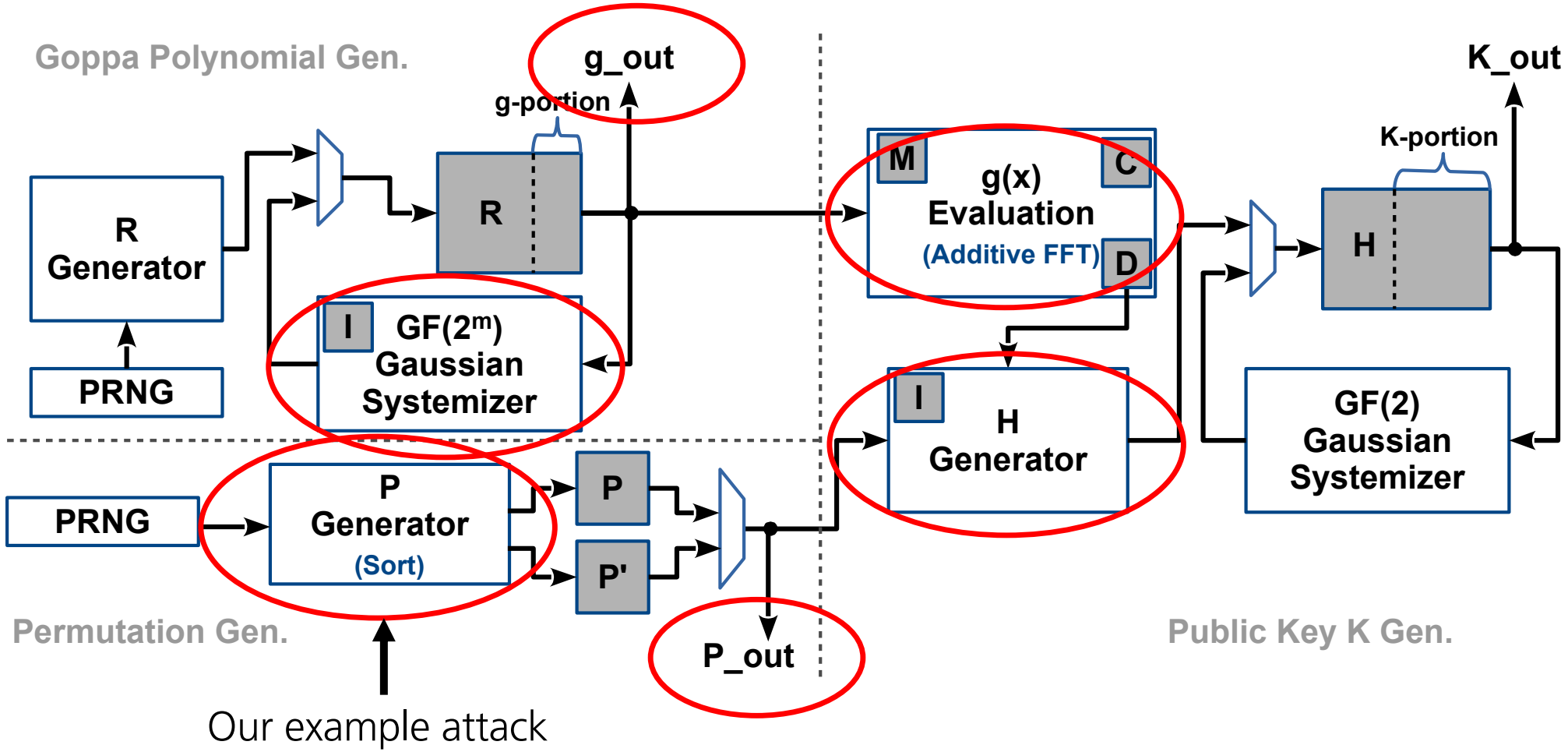
Niederreiter Cryptosystem - Analysis

Why attack the key generation?

- Private key is large
- Private key can be regenerated from a seed
- Many implementations will regenerate the private key on demand, e.g. on power up
- Power or EM trace can be recorded many times without effort (for noise reduction)

Application to Code-Based Cryptography

Niederreiter Cryptosystem - Key Generation

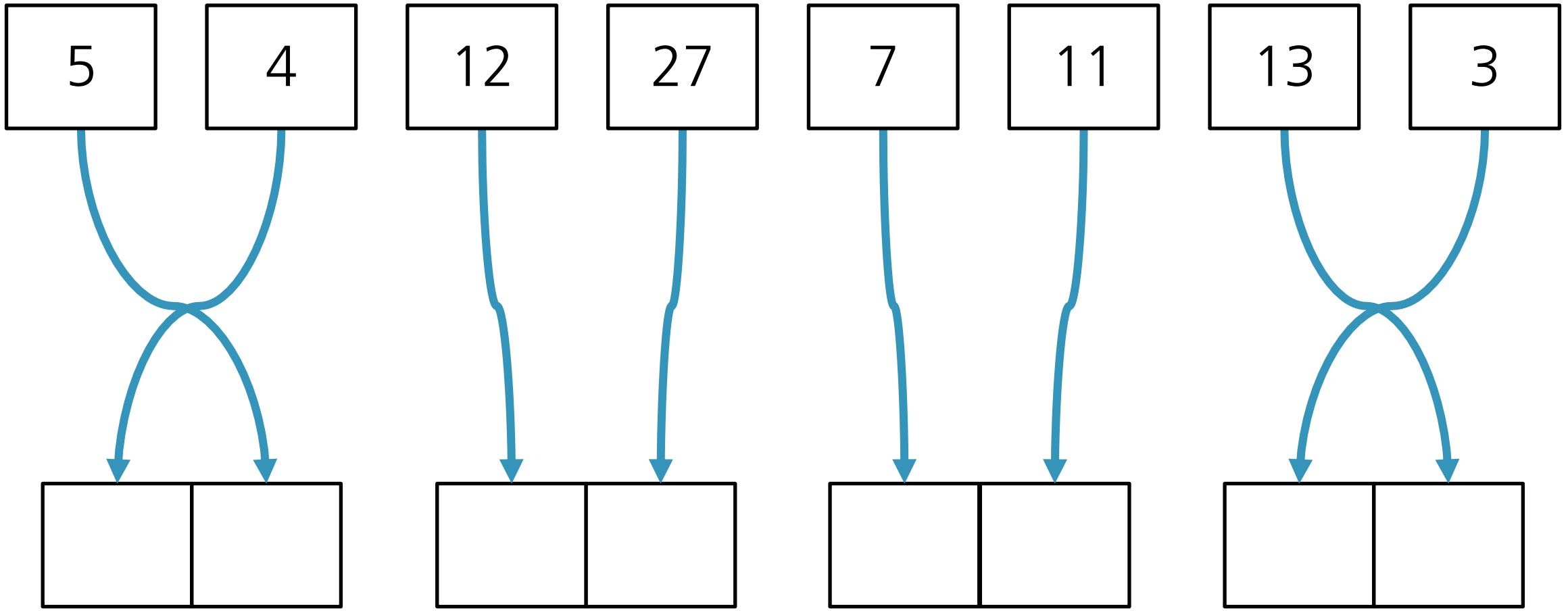


Fraunhofer SINGAPORE

Demonstration on Merge Sort

Attack on Key Generation

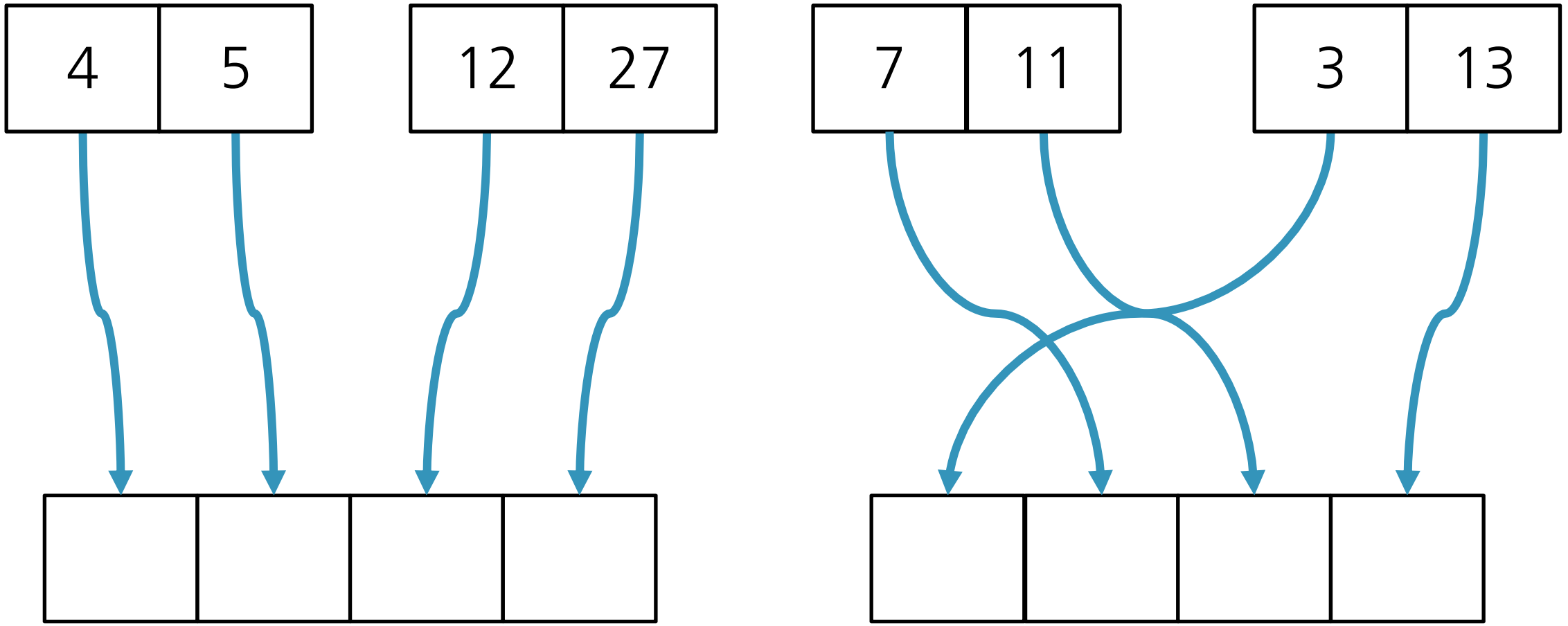
Merge Sort



Fraunhofer SINGAPORE

Attack on Key Generation

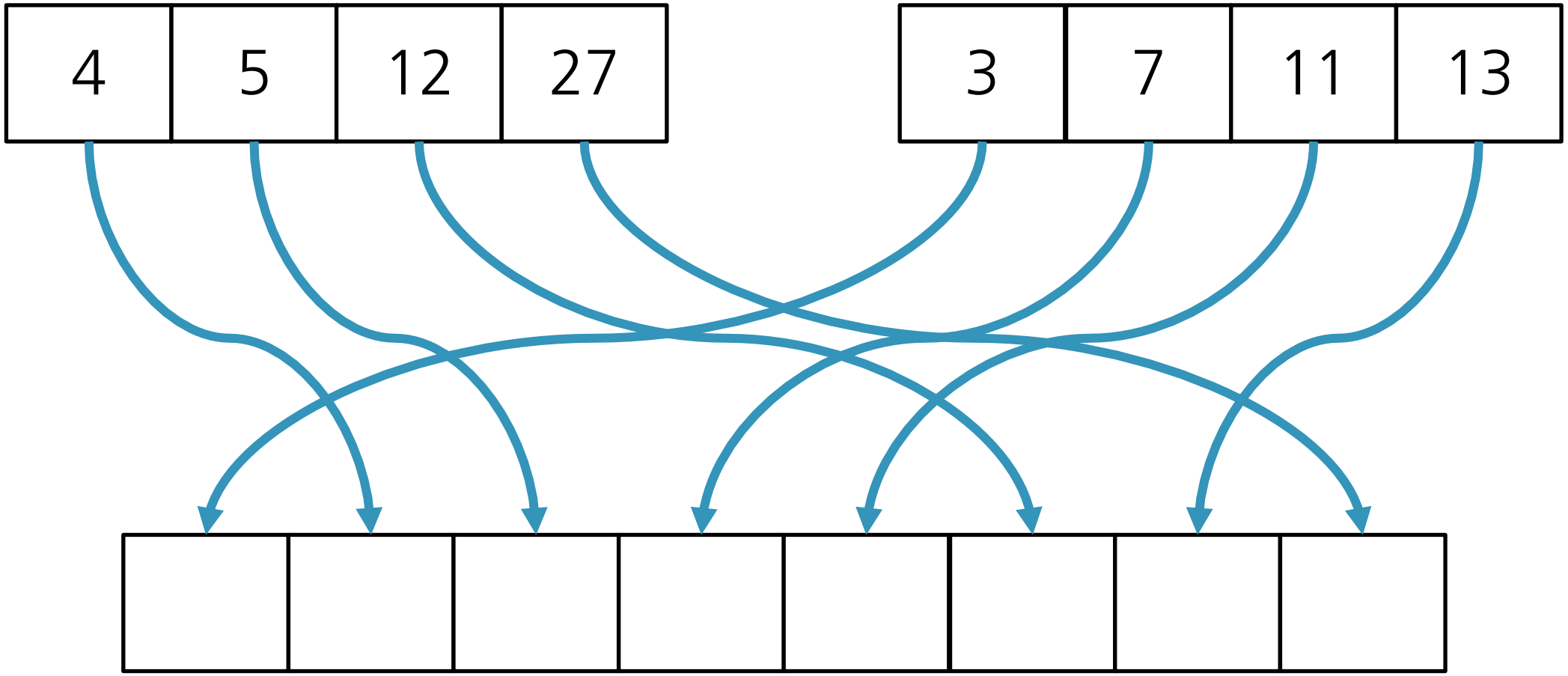
Merge Sort



Fraunhofer SINGAPORE

Attack on Key Generation

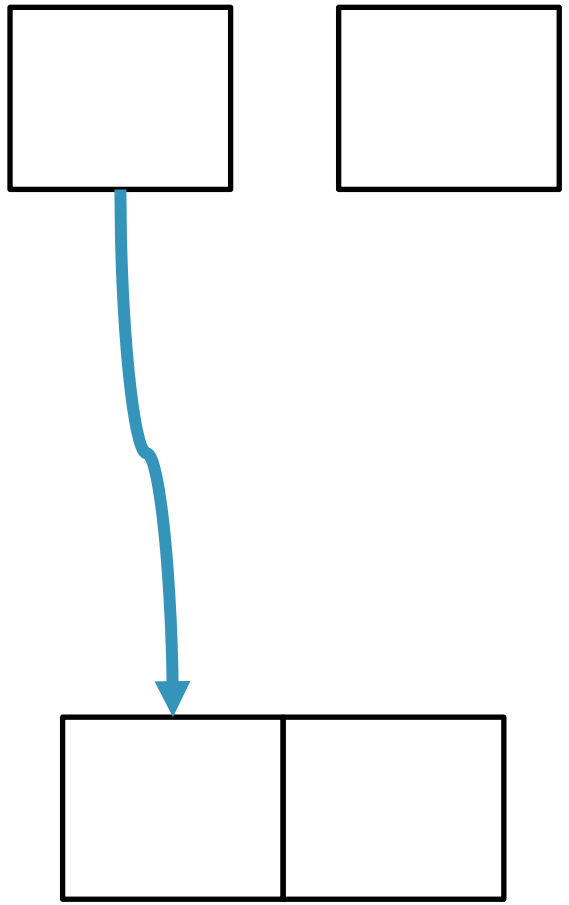
Merge Sort



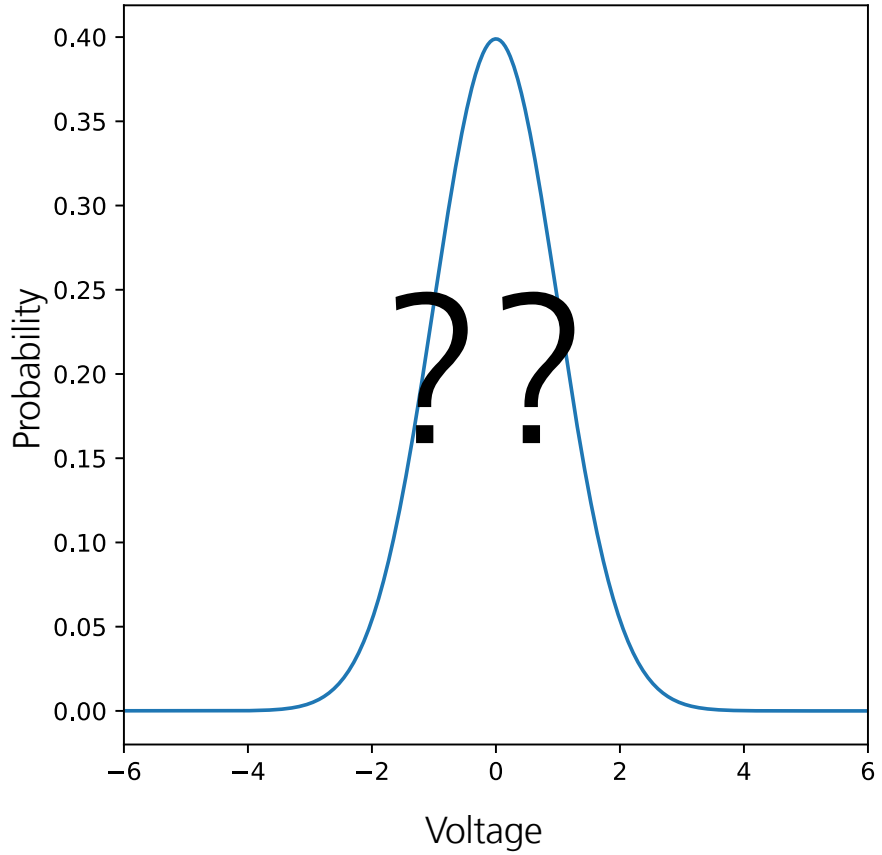
Fraunhofer SINGAPORE

Attack on Key Generation

Merge Sort – Attack Concept



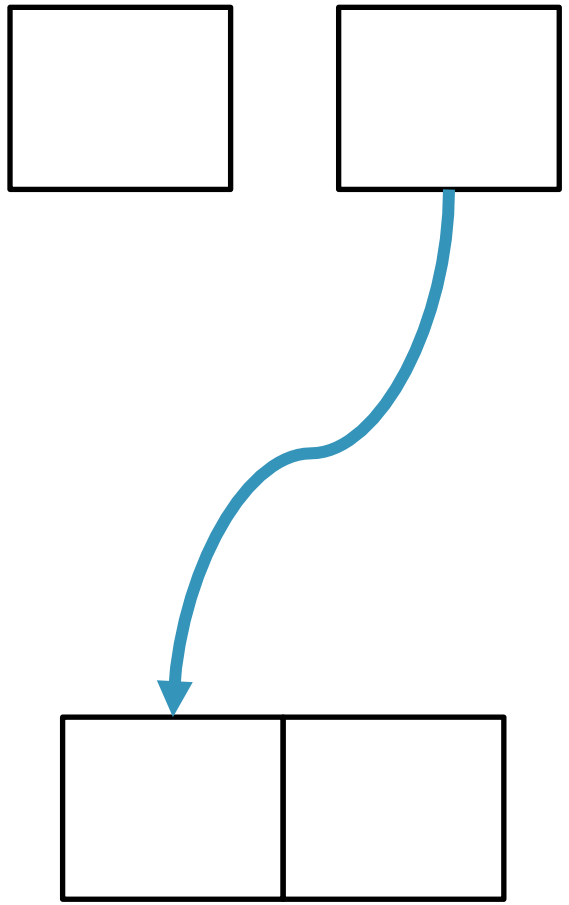
Assumption for “left” (blue)



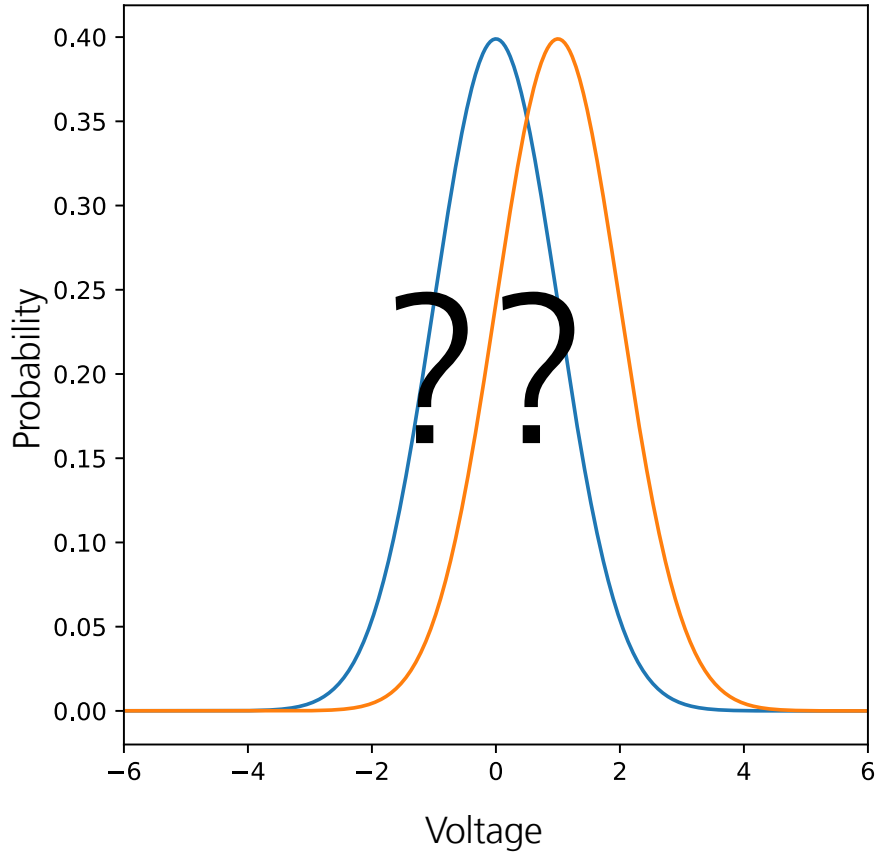
Fraunhofer SINGAPORE

Attack on Key Generation

Merge Sort – Attack Concept



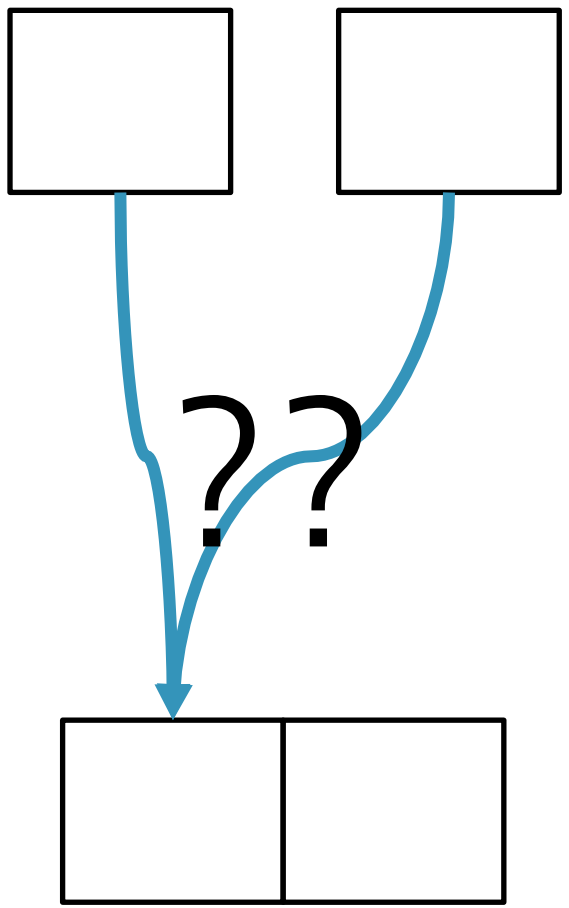
Assumption for “right” (orange)



Fraunhofer SINGAPORE

Attack on Key Generation

Merge Sort – Profiling



Remember: We are generating the same key every the time!

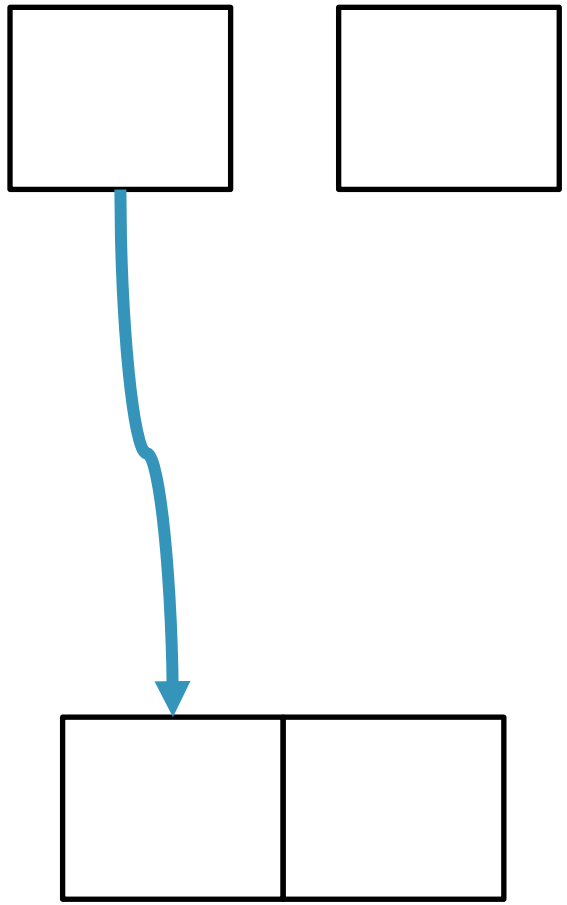
Problem: How to differentiate between “left” and “right” in a single trace?

Unknown pattern of power consumption for “left” and “right”.

Unknown position of the interesting points in a power/EM trace.

Attack on Key Generation

Merge Sort – Profiling

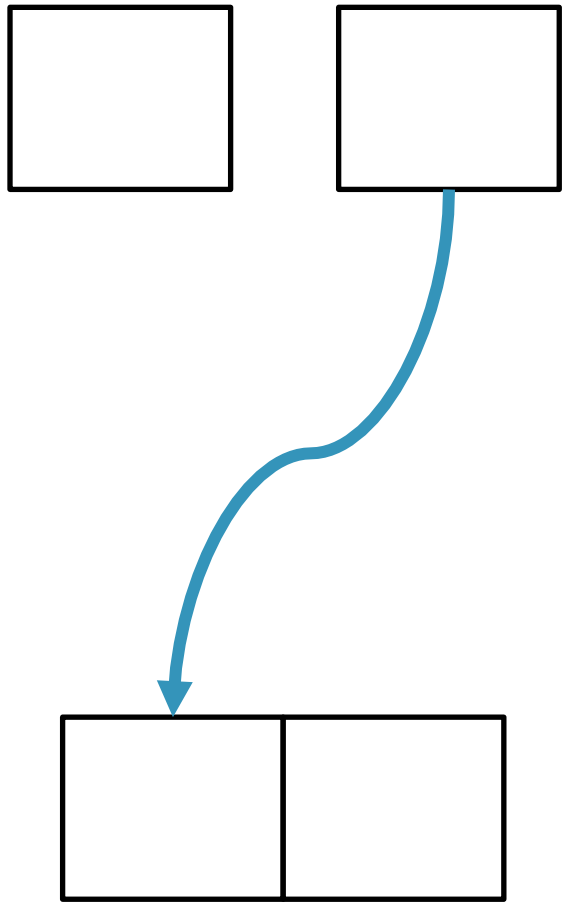


Profiling assumption:
Attacker is able to set the seed

Measure selection “left”
with different seeds.

Attack on Key Generation

Merge Sort – Profiling

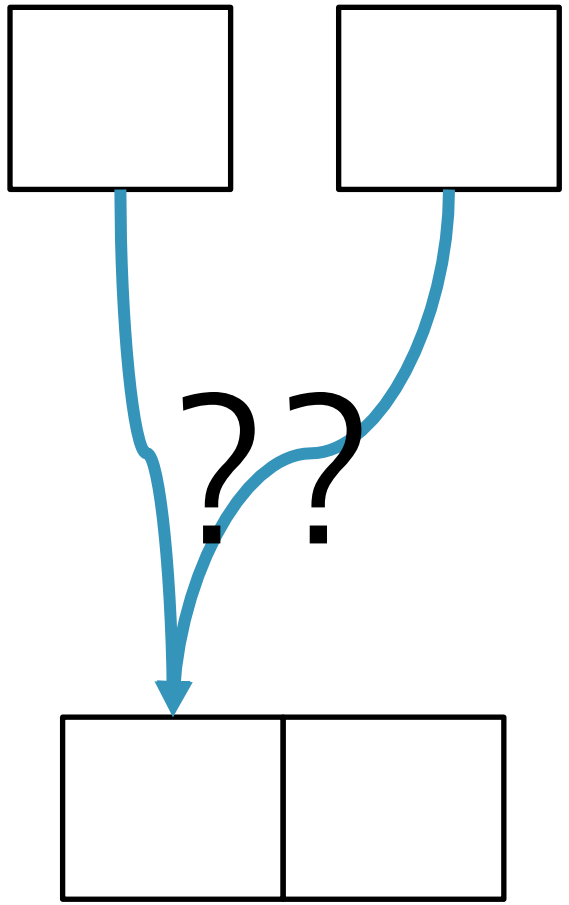


Profiling assumption:
Attacker is able to set the seed

Measure selection “right”
with different seeds.

Attack on Key Generation

Merge Sort – Profiling

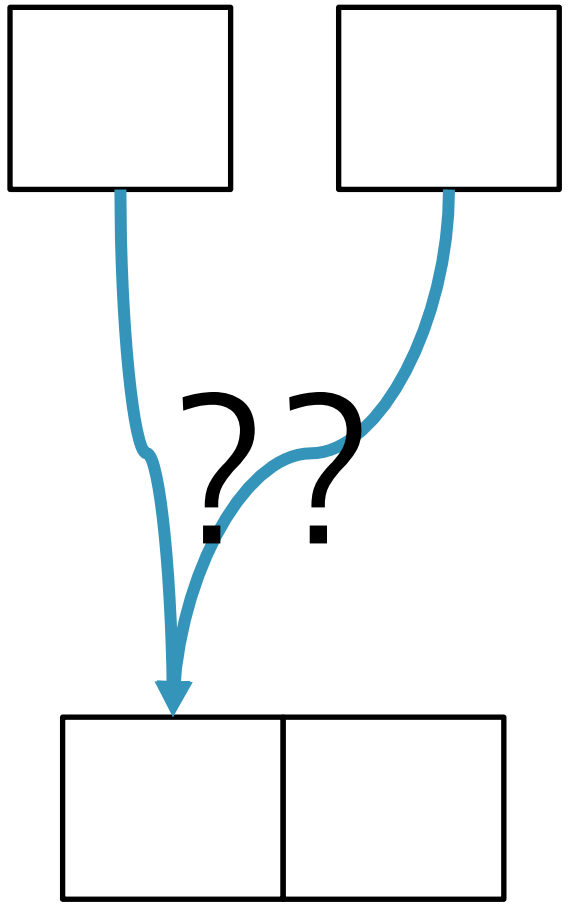


Statistical Leakage Detection:

Identifies the points of interest in the trace

Attack on Key Generation

Merge Sort – Profiling



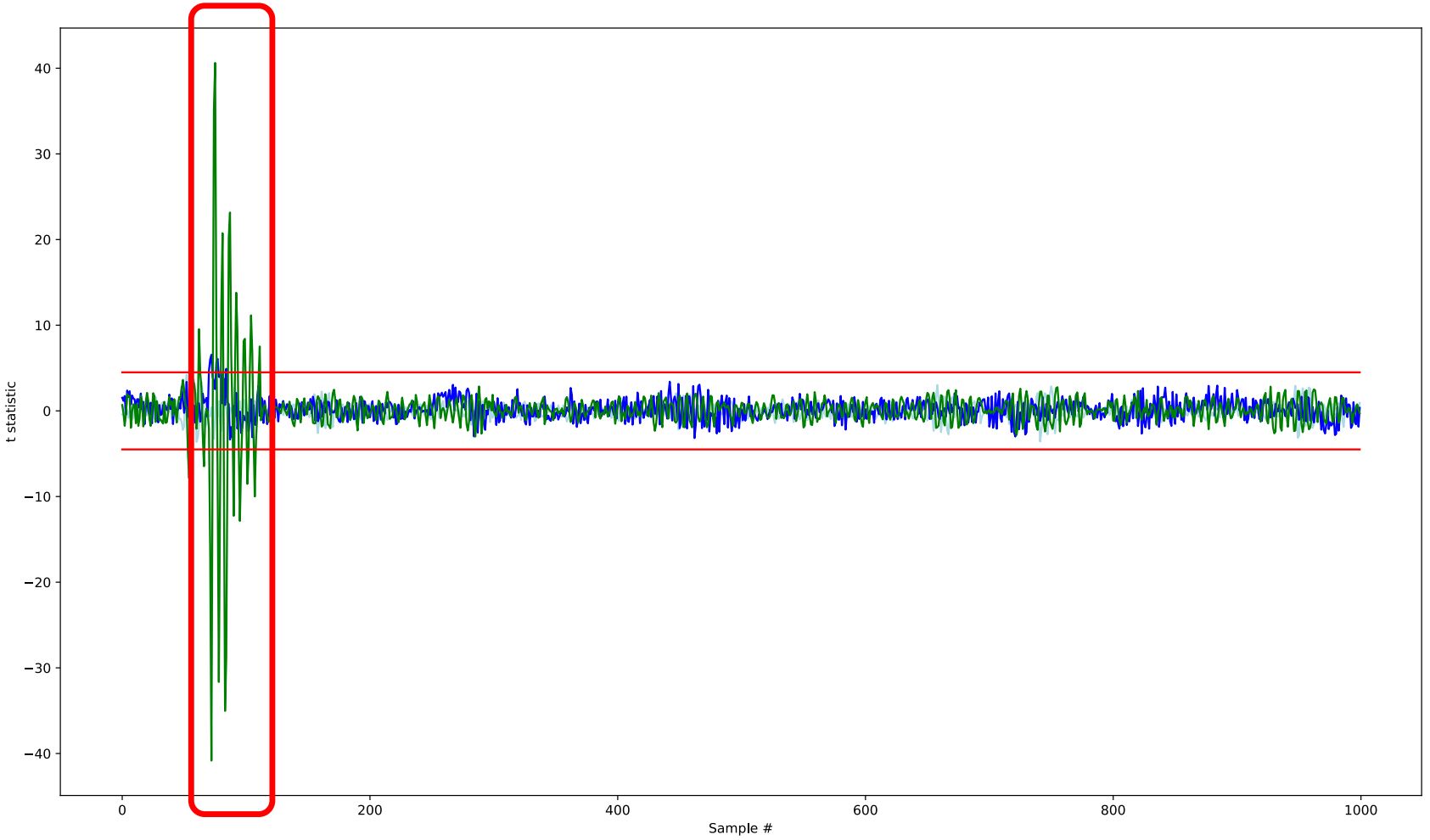
Welch's t-test:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma_1^2}{N_1} + \frac{\sigma_2^2}{N_2}}}$$

Significant leakage, when |t| > 4.5

Attack on Key Generation

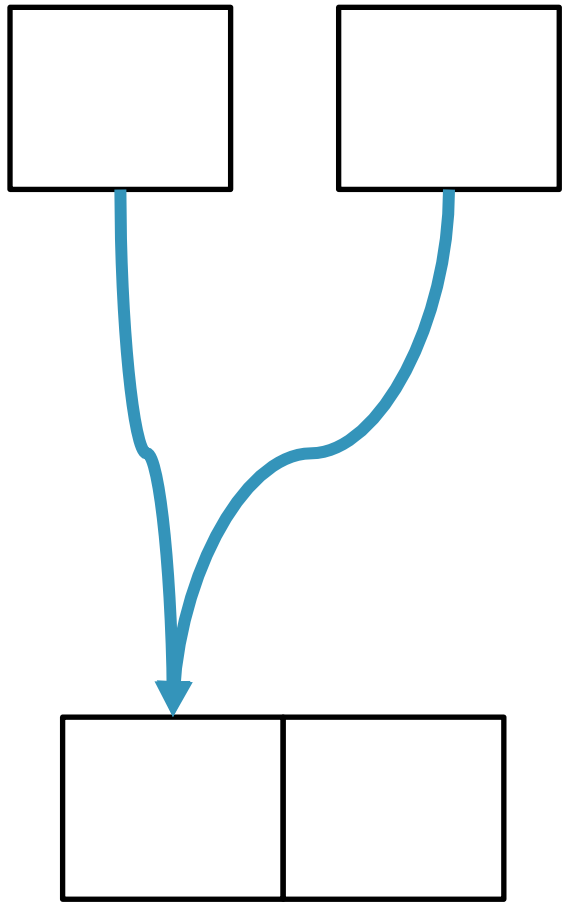
Merge Sort – Profiling



Fraunhofer SINGAPORE

Attack on Key Generation

Merge Sort – Profiling

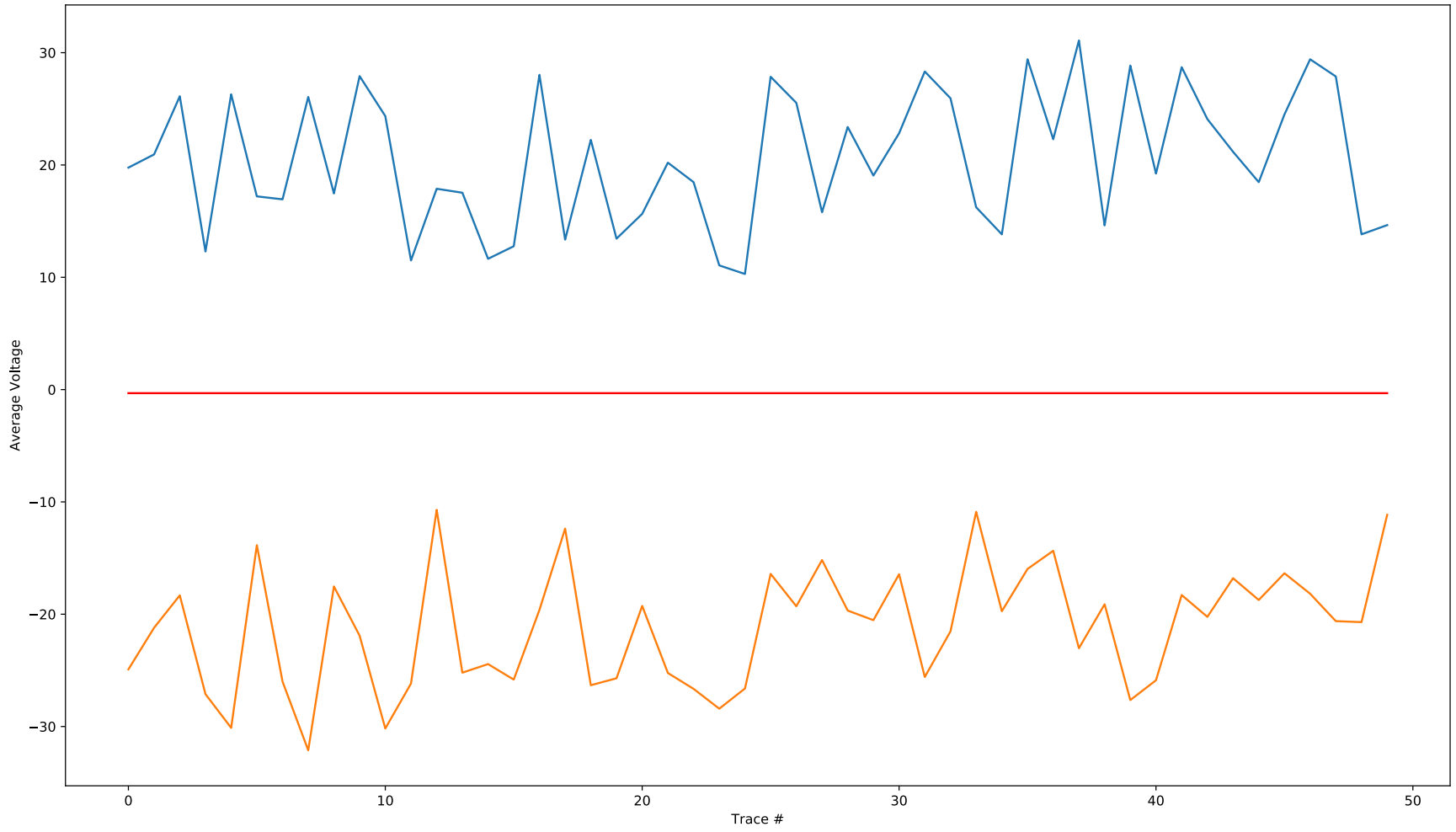


Generate Profiling Information

Simple model:
Global weighted threshold

Attack on Key Generation

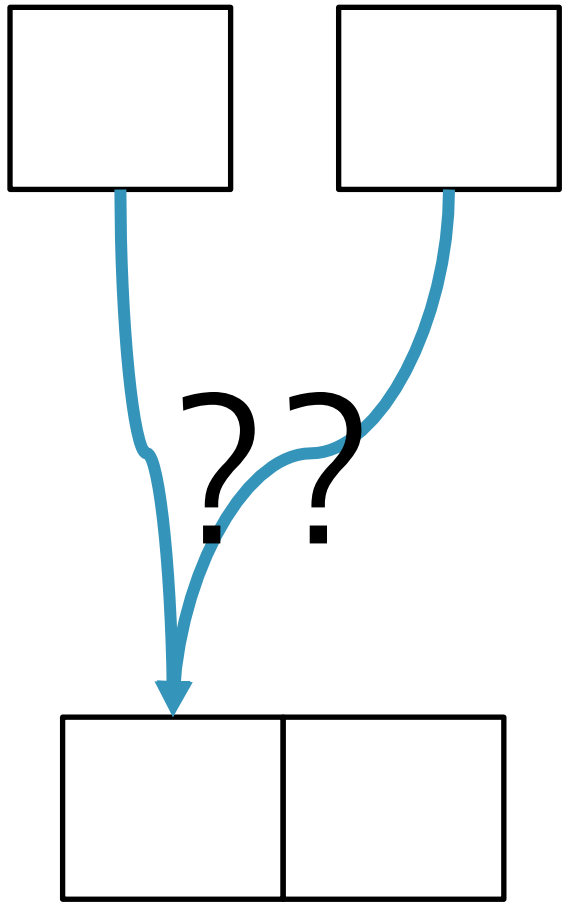
Merge Sort – Profiling



Fraunhofer SINGAPORE

Attack on Key Generation

Merge Sort – Attack Phase



Record many traces with secret key

Average traces (noise reduction)

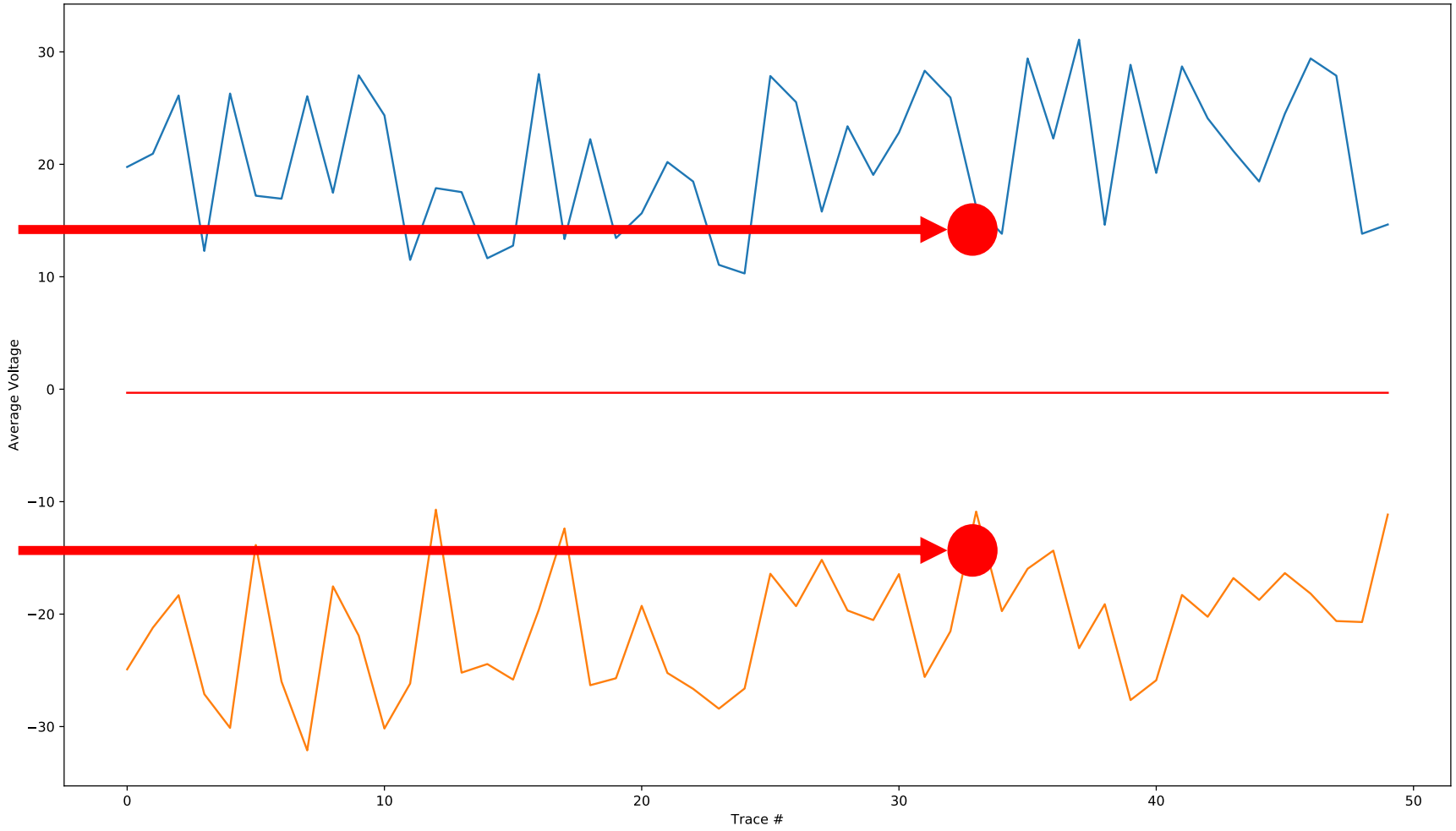
Compute weighted mean of points of interest

Attack on Key Generation

Merge Sort – Attack

"left"

"right"



Fraunhofer SINGAPORE

Countermeasures

Attack on Key Generation

Countermeasures – Hiding the Key

Randomize data?

Trivial randomization:

- Changes permutation or
- Exhibits same behavior

➔ Doesn't work!

Attack on Key Generation

Countermeasures – Hiding the Key

Randomize data?

Advanced idea:

- Generate different data
- Leading to same permutation, but
- Different “left”-”right” selection behavior

Computationally very hard ➔ Doesn’t work!

Attack on Key Generation

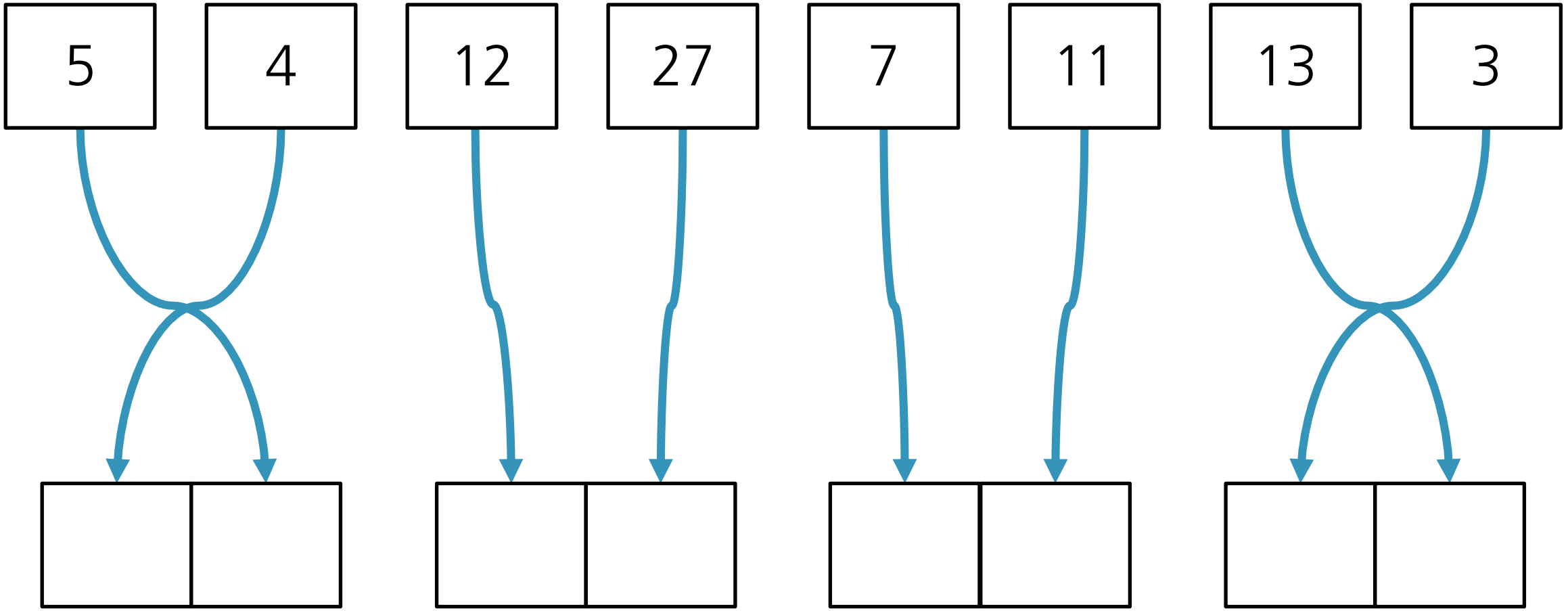
Countermeasures – Hiding the Key

Disturb signal or randomize execution

- Add clock jitter
- Add noise sources
- Shuffling of operations for each key generation

Attack on Key Generation

Countermeasures – Shuffling



Fraunhofer SINGAPORE

Attack on Key Generation

Countermeasures – Shuffling

For merge sort, there are 8192! possible permutations for the first layer alone.

SNR can be significantly reduced.

Thank you for your attention!

Fraunhofer SINGAPORE